Spatio-Temporal Deep Q-Networks for Human Activity Localization

Wanru Xu, Jian Yu, *Member, IEEE,* Zhenjiang Miao, *Member, IEEE,* Lili Wan, and Qiang Ji, *Fellow, IEEE*

***Abstract*— Human activity localization aims to recognize category labels and detect the spatio-temporal locations of activities in video sequences. Existing activity localization methods suffer from three major limitations. First, the search space is too large for three-dimensional (3D) activity localization, which requires the generation of a large number of proposals. Second, contextual relations are often ignored in these target-centered methods. Third, locating each frame independently fails to capture the temporal dynamics of human activity. To address the above issues, we propose a unified spatio-temporal deep Q-network (ST-DQN), consisting of a temporal Q-network and a spatial Q-network, to learn an optimized search strategy. Specifically, the spatial Q-network is a novel two-branch sequence-to-sequence deep Q-network, called TBSS-DQN. The network makes a sequence of decisions to search the bounding box for each frame simultaneously and accounts for temporal dependencies between neighboring frames. Additionally, the TBSS-DQN incorporates both the target branch and context branch to exploit contextual relations. The experimental results on the UCF-Sports, UCF-101, ActivityNet, JHMDB, and sub-JHMDB datasets demonstrate that our ST-DQN achieves promising localization performance with a very small number of proposals. The results also demonstrate that exploiting contextual information and temporal dependencies contributes to accurate detection of the spatio-temporal boundary.**

***Index Terms*— Activity localization, deep reinforcement learning, spatial context, temporal dependency, seq-to-seq model.**

I. Introduction

HUMAN activity analysis, which is among the most active research areas in computer vision and machine learning, has many applications, such as video content-based retrieval, human-computer interaction and video surveillance. Recently, a considerable amount of research has focused on activity recognition [44], [51], [53], which aims to assign a category label for an entire video sequence. However, activity occurs at a precise spatio-temporal extent, and it is also desirable to detect the spatio-temporal location in practical applications. For example, driving non-motor vehicles on a motorway is an abnormal activity, while driving on a non-motorway is normal. Locating human activity in the spatio-temporal domain from untrimmed videos is important and challenging, and the goal is to recognize which activity a video contains and to detect when and where the activity occurs. In recent years, most state-of-the-art activity localization methods have adopted a "proposal + classification" framework [29], [43], where the main idea is first to generate a large number of spatio-temporal candidates and then to score them to obtain the final localization result using a pretrained classifier.

Although this framework achieves promising performance, several issues exist. 1) Compared with two-dimensional (2D) object localization [37], [38], the search space is too large for 3D activity localization, especially considering a flexible bounding box size, which can vary across frames. Thus, a flexible bounding box can be utilized to detect both moving activities and nonmoving activities, while a fixed bounding box (*e.g.*, subvolume [48]) cannot handle activities with substantial motion. Therefore, an effective and efficient method to extract proposals instead of exhaustive sliding window searching must be found. 2) Most existing methods are target-centred and ignore contextual relations. However, human activity does not occur in a vacuum and context has strong relevance to the target activity. 3) Frame-based models are direct extensions of object localization methods [14] and fail to capture temporal dependencies between neighboring frames [34]. Thus, activity is independently located in each frame, and video-level localization results are obtained via temporal post-processing.

To address these three issues, we propose a unified spatio-temporal deep Q-network (ST-DQN) that is inspired by the success of DeepMind [32], [33], [50] at playing Atari games and Go. ST-DQN is a deep proposal model that aims to train a localization agent via deep reinforcement learning. Instead of time-consuming exhaustive search, the agent focuses attention on regions with rich information, following a procedure similar to human perception. The agent successively locates the activity by exploring a small number of potential time intervals and potential space regions with a temporal Q-network and a spatial Q-network.

Compared with the common temporal Q-network, the spatial Q-network is a novel two-branch sequence-to-sequence deep Q-network (TBSS-DQN). The TBSS-DQN, which takes

Manuscript received July 24, 2018; revised January 7, 2019, March 15, 2019, and April 18, 2019; accepted May 16, 2019. Date of publication May 27, 2019; date of current version September 3, 2020. This work was supported in part by the NSFC under Grant 61672089, Grant 61273274, and Grant 61572064, in part by the China Postdoctoral Science Foundation under Grant 2019M650469, and in part by the National Key Technology R&D Program of China under Grant 2012BAH01F03. This paper was recommended by Associate Editor H. Lu. *(Corresponding author: Wanru Xu.)*

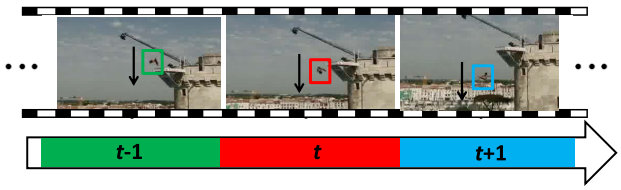
W. Xu, Z. Miao, and L. Wan are with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China (e-mail: xuwanru@bjtu.edu.cn; zjmiao@bjtu.edu.cn; llwan@bjtu.edu.cn).

J. Yu is with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China (e-mail: jianyu@bjtu.edu.cn).

Q. Ji is with the Department of Electrical & Computer Engineering, Rensselaer Polytechnic Institute, NY 12180 USA (e-mail: qji@ecse.rpi.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2019.2919064



(a)

(b)

Fig. 1. (a) Temporal dependency: it tends to select consistent actions between neighboring frames. In the activity of 'diving', the changing trend of the bounding box is downward motion for each frame. (b) Contextual relation: inaccurate localization can not only affect the activity itself (target) but also negatively affect its context. The left image shows accurate localization, and the right image is inaccurate.

both contextual relations and temporal dependencies into account, has two advantages. First, TBSS-DQN is a sequence-to-sequence model that makes a sequence of decisions to change the attentional location of each frame simultaneously. The sequential Q-network successfully incorporates the global temporal dependencies, where each change in decision sequence is made depending on both the state of the current frame and the cues from previous frames. Fig. 1(a) shows a temporal dependency, namely, the network tends to select consistent action between neighboring frames. Second, TBSS-DQN is a two-branch model that associates a target activity with its context to model the underlying relationship. Recent works [15], [17] have shown that incorporating contextual information leads to a significant improvement in activity recognition. Context is important not only for activity recognition but also for activity localization because the boundary is just the intersection between the target activity itself and its context, which inevitably influence each other, as shown in Fig. 1(b). Specifically, if the detected bounding box is smaller than the ground truth, part of the target is lost, and this lost region will be mistakenly considered as part of the context. In experiments, we show that our proposed model is capable of effective reasoning reasoning regarding temporal bounds and spatial bounding boxes. More importantly, the ST-DQN achieves promising performance with a very small number of proposals. To the best of our knowledge, this is the first deep reinforcement learning model for human activity spatio-temporal localization.

Overall, the contributions of this paper are as follows: 1) Human activity spatial localization and temporal localization are addressed simultaneously with a unified ST-DQN, which is an effective and efficient way to extract proposals via an optimized attention strategy instead of exhaustive search. 2) A novel TBSS-DQN is proposed to make a sequence of decisions to locate each frame simultaneously and to integrate the context into the interpretation process to provide double cues for activity localization. The TBSS-DQN fully captures dynamic temporal relationships between neighboring frames and exploits contextual information in the area surrounding the target activity. 3) The experiment demonstrates that our spatio-temporal deep Q-network is a good proposal model that can generate high-quality proposals for human activities, namely, the network achieves precise results using only a very small number of proposals.

II. Related Work

A. Proposal + Classification Based Localization Methods

Currently, human activity localization is commonly approached by spatio-temporal proposals matching [3], [10], [36], namely, the classic proposal + classification framework [56], [64], [70]. Traditionally, proposals are generated by a sliding window [12], [26], which is effective but not efficient. Several efforts aim to reduce the computational cost, such as the spatio-temporal branch-and-bound algorithm [66]. The activity proposal model is another alternative to reduce the search space considerably. The 2D deformable part model (DPM) is extended to a 3D spatio-temporal DPM in [48], where the most discriminative 3D subvolumes are selected as individual proposals. The naive Bayes-based mutual information maximization (NBMIM) [65] method has been introduced to find the optimal subvolume in 3D video space to efficiently detect activity. In addition to these subvolume-based methods, which cannot detect moving activities, spatio-temporal tubes are extracted via structured output regression [49] with a max-path search. A series of spatio-temporal video tubes [63] are considered as localization candidates and are generated via a greedy method by computing an actionness score. Activity proposals are defined as 2D+t sequences of bounding boxes called tubelets in [20], which are generated by hierarchically merging supervoxels. Although these proposal models achieve promising performance, they are still time-costuming and do not follow the human perception procedure of searching.

Many frame-based methods [23], which are direct extensions of object localization approaches, have been proposed for human activity localization. Poselet is extended to a dynamic poselet model [52], where human activity is first decomposed into temporal "key poses" and then further decomposed into spatial "action parts". Due to the excellent performance for object detection, many works prefer to use convolutional neural networks (CNNs) to extract proposals for each frame and then either link or track the proposals to obtain the final video-based localization result. In [58], frame-level proposals are first detected and then scored using a combination of static and motion CNN features. Next, the proposals with high scores are tracked in the video sequence with a tracking-by-detection approach. A multiregion two-stream R-CNN model [34] is introduced to locate activity in realistic videos by starting from frame-level detection based on faster R-CNN [14] and then linking frame-level detections to obtain video-level detections. In [45], a single-shot multi-box detector (SSD) is adopted to regress the bounding box in each frame; then, action tubes are constructed from these SSD frame-level detections. In the

tube CNN [8], a set of tube proposals are generated depending on 3D convolutional network features for each equal-length clip; then, tube proposals from different clips are linked by network flow. Overall, such indirect methods, which ignore the temporal dynamics of human activity, are unsatisfying in terms of search efficiency and localization accuracy.

B. Contextual Models for Activity Recognition

Recently, the advantage of combining context for human activity recognition, which can significantly improve the recognition performance, has been demonstrated by many works. Various contextual elements have been integrated as well, including spatial context [15], [17] and temporal context [35], [69]. An r∗CNN [15] has been introduced to use more than one region to construct a strong activity recognition system. Additionally, a context-augmented event recognition approach [57] has been proposed to capture three levels of context from time to space, namely, image level, semantic level, and prior level. In [35], a temporal embedding is learned for complex video analysis by associating frames with the temporal context. A novel active learning technique, which not only exploits evidence from individual activity instances but also utilizes contextual information among activities and objects, is formulated in [17]. A framework of collective human activity recognition that automatically captures the relevant context of a crowd with a 3D Markov random field is proposed in [41]. In [1], a novel deep action- and context-aware sequence learning, which effectively combines both context-aware and action-aware features via a multistage recurrent architecture, is presented for activity recognition and anticipation. A two-graph model [60] is constructed to represent human activities by modeling spatial and temporal relationships among local features. Then, a novel family of context-dependent graph kernels (CGKs) is proposed to measure the similarity between graphs for matching activities. A recurrent interactional context model, which is an extension of long short-term memory (LSTM) network, is proposed to capture high-order interactional context in [54]. Meanwhile, a unified interactional feature modeling process is introduced for one-person dynamics and for intragroup and intergroup interactions. However, most current activity localization approaches are still target-centred and give insufficient attention to contextual information.

C. Deep Reinforcement Learning in Computer Vision

In recent years, an increasing number of computer visual problems, such as object detection [2], [5], [22], recognition [6] and tracking [9], [67], [68], have been formulated in the deep reinforcement learning framework. A fully end-to-end approach for temporal activity localization is presented in [62], where the agent is trained by reinforcement learning, which directly learns to predict temporal bounds of activities. An active model for localizing objects is proposed in [5], where an agent is allowed to focus attention on candidate regions to find the location of the target object accurately and rapidly. An effective tree-structured reinforcement learning (Tree-RL) method [22] has been introduced to sequentially locate objects by fully exploiting both historical search paths and current observations. Similarly, objects are located by executing hierarchical object detection in 2D images guided by a deep reinforcement learning agent in [2]. A collaborative deep reinforcement learning method is proposed for joint multiple objects detection by treating each detector as an agent in [25]. The researchers utilize a novel multi-agent deep Q-learning algorithm to learn inter-agent communication, which effectively exploits beneficial contextual information. In [68], a novel neural network tracking model is proposed; the model comprises three sub-models: a CNN for extracting features from each frame, a recurrent neural network (RNN) for constructing temporal states, and a reinforcement learning agent for making decisions to locate a target. A template selection strategy constructed by deep reinforcement learning for visual tracking, which utilizes this strategy to select the best template for tracking a certain frame in videos, is introduced in [9]. In [67], an action-decision network (ADNet), which is trained by supervised learning and deep reinforcement learning, is applied to control a novel tracker by sequentially pursuing actions. In [6], an attention-aware deep reinforcement learning (ADRL) method, which aims to discard some misleading frames and find the most informative frames to represent face videos, is introduced for video-based face recognition.

In general, deep reinforcement learning provides a coarse-to-fine search strategy, where observation and refinement are performed iteratively. Such detection methods are similar to the process of human perception, so they do not require a large number of proposals. However, these methods are used either to train a spatial localization agent for object detection or to train a temporal localization agent for activity temporal detection. In contrast to the existing deep reinforcement learning networks, the proposed ST-DQN is designed to achieve a light computational burden and satisfactory localization accuracy in both the spatial and temporal domains. A novel two-branch sequence-to-sequence deep Q-network is proposed to incorporate the temporal dynamics and contextual information of human activities to improve the extension of frame-level detection to video-level detection.

III. The ST-DQN for Activity Localization

The goal of human activity localization is to detect the exact activity location in a video sequence, which is defined as a series of bounding boxes $\mathbf{B} = \{x_1^t, y_1^t, x_2^t, y_2^t\}_{t_1}^{t_2}$ between the start frame t_1 and end frame t_2. Each bounding box is represented by the coordinates of two of its corners. In this paper, we propose a unified ST-DQN for activity localization, consisting of a temporal Q-network and a spatial Q-network, as depicted in Fig.2. The temporal model is the common Q-network used for searching the time period $\{t_1, t_2\}$. The spatial model is a novel TBSS-DQN that makes a sequence of decisions to search the bounding box for each frame simultaneously. Note that TBSS-DQN is a generalized sequence-to-sequence model that is not based on the typical encoder-decoder structure. Thus, when given input sequential video frames, TBSS-DQN decodes each output for each frame

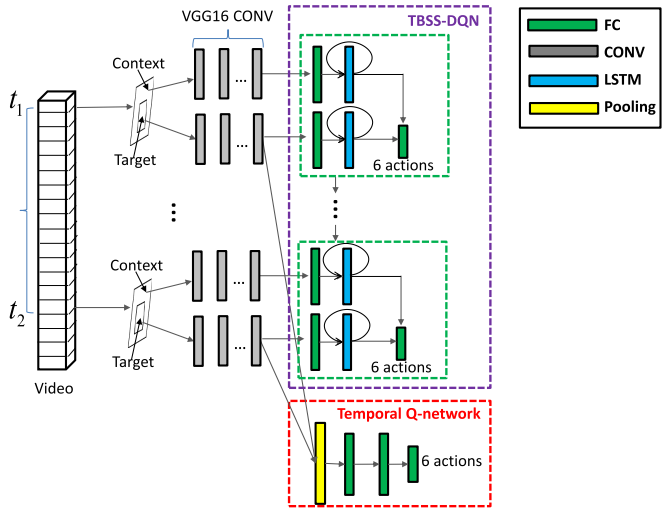


Fig. 2. Illustration of the spatio-temporal deep Q-network (ST-DQN), consisting of a temporal Q-network and a spatial network (TBSS-DQN), for human activity localization. Before being fed to the ST-DQN, each frame of input video is warped to 224*224 pixels. Then, the frames are processed by a pretrained CNN following the VGG-16 architecture. The output of the CNN is concatenated as the sequential state representation, which is taken as the input of the spatial Q-network. The TBSS-DQN incorporates the target branch and context branch to predict a sequence of spatial actions, where we take the feature fusion as an example here. The input of the temporal Q-network is calculated by temporal pooling via averaging of all the CNN outputs, and the output is the predicted temporal actions.

from all the previous frames and current frame, and it then outputs sequential decisions frame by frame. The TBSS-DQN not only accounts for temporal dependencies between neighboring frames but also incorporates the target branch and context branch to take full advantage of contextual relationships.

We cast the problem of human activity localization as a Markov decision process (MDP) to rapidly find the spatio-temporal location of an activity by a trained localization agent. A video sequence is considered as the environment, where the agent sequentially deforms a time interval and a series of bounding boxes using a set of predefined actions in the temporal and spatial domains. Typically, an MDP is defined as an (S, A, R) tuple, namely, a set of states $s \in S$, a set of actions $a \in A$, and a reward function $R(s, a)$. At each time step, the localization agent estimates the current state s and takes an optimal action a^* to find the next better location, $a^* = \arg\max_a Q(s, a)$, which is considered as a policy mapping from the state set to the action set. The optimal action is determined by maximizing the Q-function, which is approximated by the spatial or temporal Q-network.

A. The Two-Branch Sequence-to-Sequence Deep Q-network

For human activity spatial localization, we propose a novel two-branch sequence-to-sequence deep Q-network (TBSS-DQN) that fully models both contextual information and temporal dynamics. Therefore, each element of the MDP in TBSS-DQN is in a sequential form, including the spatial state $\mathbf{s}^S = \{\mathbf{s}_1, \mathbf{s}_2\}$, spatial action $\mathbf{a}^S = \{\mathbf{a}_1, \mathbf{a}_2\}$ and spatial reward $R(\mathbf{s}^S, \mathbf{a}^S)$ for the target and context components.

1) Network Architecture: The Q-function is defined as an action-value function that accumulates the expected total future discounted reward:

$$Q^S(\mathbf{s}^S, \mathbf{a}^S) = E[\sum_{k=m}^{K} \gamma^{k-m} R(\mathbf{s}_k, \mathbf{a}_k) | \mathbf{s}_m = \mathbf{s}^S, \mathbf{a}_m = \mathbf{a}^S; \theta^S] \quad (1)$$

Instead of the currently returned reward $R(\mathbf{s}_k, \mathbf{a}_k)$, the agent takes all the future possibilities into account for action selection. As shown in Eq.(1), the spatial Q-function reflects the improvement in localization accuracy during the whole running episode K, where $\mathbf{s}_m \rightarrow \mathbf{s}_k$ is a state transition after $k - m$ steps and γ is a discount factor.

Due to the high-dimensional continuous state and model-free environment, the problem is difficult to solve via traditional reinforcement learning. Therefore, we learn the search strategy by considering TBSS-DQN as an approximator of this spatial Q-function to estimate the optimal value for each state-action pair, as shown in Fig.2. Technically, for TBSS-DQN, given an activity video $\mathbf{I} = \{I_1, I_2, \ldots, I_T\}$, the corresponding optimal action of $t-th$ frame at $k-th$ searching step is computed by

$$\begin{aligned} s_{k,t} &= \phi_{VGG-16}(I_t, bb_{k,t}) \\ h_{k,t} &= \phi_{LSTM}(s_{k,t}, h_{k,t-1}) \\ q_{k,t} &= \phi_{RL}(h_{k,t}) \\ p_{k,t} &= \phi_{soft\max}(q_{k,t}) \\ a_{k,t} &= \arg\max_a q_{k,t}(s_{k,t}, a) \end{aligned} \quad (2)$$

First, a CNN is utilized to encode the visual information of frame I_t within bounding box $bb_{k,t}$, which is called the current state $s_{k,t}$. This current state, together with the previous hidden state, is fed into the $t-th$ input node of LSTM [19] to drive the state transition from $h_{k,t-1}$ to $h_{k,t}$. The q-value $q_{k,t}$ is calculated based on the hidden state $h_{k,t}$, which includes all available information. Finally, we obtain output probability $p_{k,t}$ by soft-max normalization and select the action with the largest q-value as the optimal action $a_{k,t}$ to change the bounding box $bb_{k,t}$. Collectively, ϕ_{VGG-16}, ϕ_{LSTM} and ϕ_{RL} constitute the TBSS-DQN parameterized by θ^S.

For the TBSS-DQN, we combine the target branch and context branch to obtain stronger evidence for accurate localization. Each branch in the TBSS-DQN consists of a fully connected (FC) layer of 1024 neurons, an LSTM [19] layer of 1024 neurons, and an output layer with a sequence of 6 action decisions. Given the two branches $\mathbf{s}_k^1 \rightarrow \mathbf{q}_k^1$ and $\mathbf{s}_k^2 \rightarrow \mathbf{q}_k^2$, following Eq.2, we want to learn a unified $\mathbf{s}_k^S \rightarrow \mathbf{q}_k^S$. As shown in Fig.3, we consider four different architectures to fuse the two branches, including decision fusion, feature fusion, unidirectional expert and bidirectional expert. Note that slight differences exist between the four fusion strategies, which we detail as follows: (1) Decision fusion in Fig. 3(a): This method integrates two branches in terms of the final q-value by a summation operation:

$$\mathbf{q}_k^S = \mathbf{q}_k^1 + \mathbf{q}_k^2 = \{q_{k,1}^1 + q_{k,1}^2, \ldots, q_{k,T}^1 + q_{k,T}^2\} \quad (3)$$

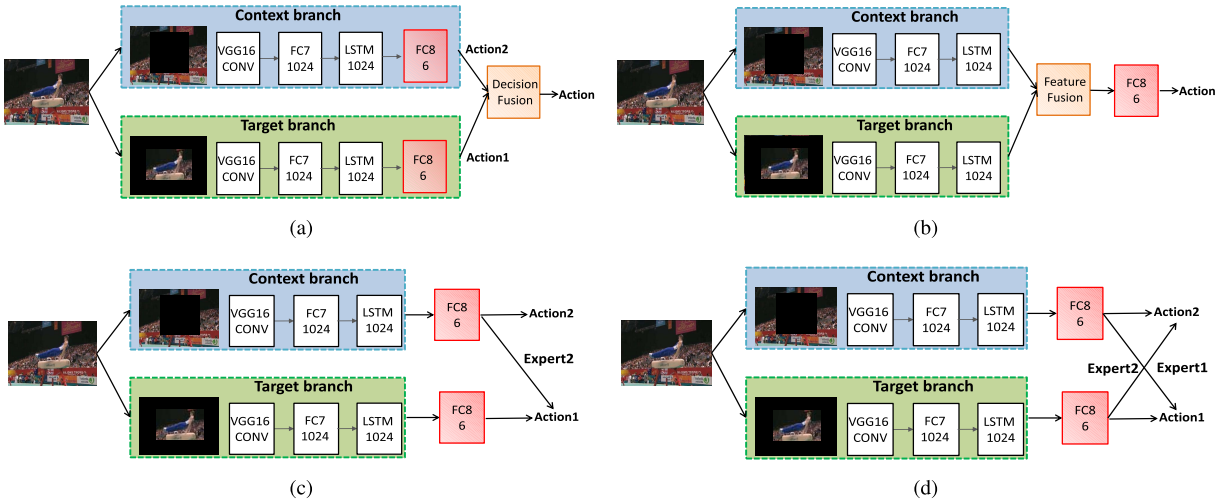


Fig. 3. Illustration of our proposed TBSS-DQN for human activity spatial localization. Note that although the network is a sequential model that simultaneously processes multiple frames in a video, we present only one frame for simplicity. (a)(b) TBSS-DQN with decision fusion/ feature fusion: the target network and context network are integrated by decision fusion or feature fusion. (c)TBSS-DQN with unidirectional expert: the context branch is treated as an expert to refine the detection result of the target branch. (d)TBSS-DQN with bidirectional experts: the two branches support and complement each other.

The two branches are trained independently, and their loss functions depend on the Bellman equation [32]:

$$L^1 = \sum_{t,k}[\gamma \max_{a^1_{k+1,t}} Q^1(s^1_{k+1,t}, a^1_{k+1,t}; \theta^1) + R(s^1_{k,t}, a^1_{k,t}) - Q^1(s^1_{k,t}, a^1_{k,t}; \theta^1)]^2 \quad (4)$$

$$L^2 = \sum_{t,k}[\gamma \max_{a^2_{k+1,t}} Q^2(s^2_{k+1,t}, a^2_{k+1,t}; \theta^2) + R(s^2_{k,t}, a^2_{k,t}) - Q^2(s^2_{k,t}, a^2_{k,t}; \theta^2)]^2 \quad (5)$$

(2) Feature fusion in Fig. 3(b): To consider the pixel-wise correspondences between the target branch and context branch, we fuse the two branches in the feature layer and then directly feed it to FC8 to compute $\mathbf{s}^S_k \rightarrow \mathbf{q}^S_k$:

$$\mathbf{s}^S_k = \mathbf{s}^1_k + \mathbf{s}^2_k = \{s^1_{k,1} + s^2_{k,1}, \ldots, s^1_{k,T} + s^2_{k,T}\} \quad (6)$$

In addition to the above two traditional fusion approaches, we propose two novel fusion methods that are more adaptive in the reinforcement learning framework. (3) Unidirectional expert in Fig. 3(c): To train the target branch, we consider the context branch as an additional expert. The loss function of the context branch is still computed by Eq. 5, but that of the target branch is converted to the following:

$$L^1 = \sum_{t,k}[R(s^1_{k,t}, a^1_{k,t}) - Q^1(s^1_{k,t}, a^1_{k,t}; \theta^1) + \gamma \max_{a^1_{k+1,t}} Q^1(s^1_{k+1,t}, a^1_{k+1,t}; \theta^1)]^2 - p^2_{k,t} \log p^1_{k,t} \quad (7)$$

In Eq. 7, the context branch provides supervised information $p^2_{k,t}$ for training the target branch by a minimum cross entropy constraint, which requires the output probabilities of the two branches to be similar. (4) Bidirectional expert in Fig. 3(d): In contrast to unidirectional fusion, bidirectional fusion requires the two branches to act as experts for each other. The training of the context branch is also supervised by the target branch:

$$L^2 = \sum_{t,k}[R(s^2_{k,t}, a^2_{k,t}) - Q^2(s^2_{k,t}, a^2_{k,t}; \theta^2) + \gamma \max_{a^2_{k+1,t}} Q^2(s^2_{k+1,t}, a^2_{k+1,t}; \theta^1)]^2 - p^1_{k,t} \log p^2_{k,t} \quad (8)$$

For the last two fusion strategies, $\mathbf{q}^S_k = \mathbf{q}^1_k$ is obtained by $\mathbf{s}^1_k \rightarrow \mathbf{q}^1_k$. The context branch is used as an expert to help the target branch make better decisions, since there is no ground truth for each change in the bounding box during training.

2) Spatial State: The spatial localization agent has a state representation with information of the currently visible region, composed by the descriptors inside the bounding box and outside the bounding box. The former is used for the target branch, while the latter is taken as the input of the context branch. The target branch and context branch provide double cues for accurate localization and complement each other. Before being fed to the TBSS-DQN, each input video frame is first warped to 224*224 pixels. As shown in Fig.2, the state of the target branch is a sequence of representations $\mathbf{s}^1_k = \{s^1_{k,1}, s^1_{k,2}, \ldots, s^1_{k,T}\}$ extracted from the inside of the current region at the $k-th$ searching step using a pretrained CNN following the VGG-16 architecture. We introduce a model with the same architecture to extract the state of the context branch, which represents the outside of the current region at the k–th searching step by masking the target regions, denoted as $\mathbf{s}^2_k = \{s^2_{k,1}, s^2_{k,2}, \ldots, s^2_{k,T}\}$.

3) Spatial Action: Two different spatial action sets are available for changing the bounding boxes, which would be evaluated and compared in the experimental part. The first set of actions for spatial localization is composed of 5 transformation actions (top-left scaling, top-right scaling, bottom-left scaling, bottom-right scaling, and centre scaling) and one terminate action. For example, centre scaling with a scaling

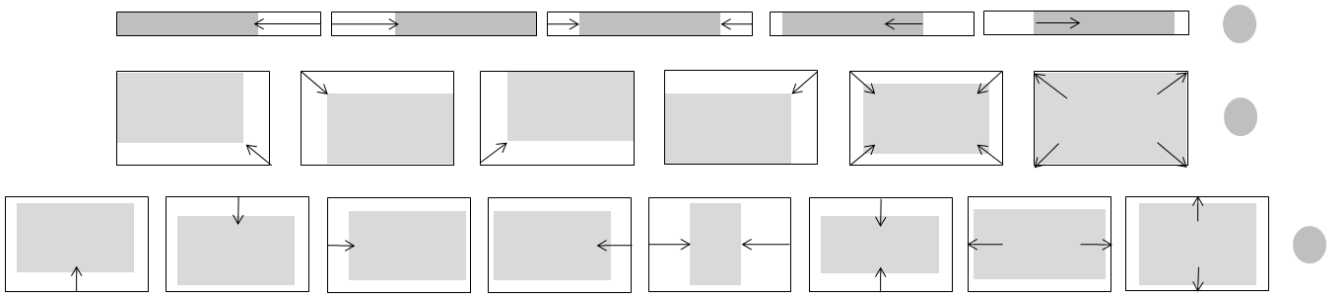


Fig. 4. Illustration of the action set in the proposed ST-DQN. (Top) 6 actions for temporal localization; (Middle) 6 actions for spatial localization; (Bottom) 9 actions for spatial localization.

factor α is performed in the following way:

$$x_1 = x_1 + \alpha \times (x_2 - x_1)/2, \quad y_1 = y_1 + \alpha \times (y_2 - y_1)/2 \quad (9)$$

$$x_2 = x_2 - \alpha \times (x_2 - x_1)/2, \quad y_2 = y_2 - \alpha \times (y_2 - y_1)/2 \quad (10)$$

To stop the searching process, we introduce the terminate action, which indicates that the agent has reached the location of the activity of interest. The second set of actions for spatial localization is composed of 8 transformation actions (moving left/right, moving up/down, becoming shorter/longer horizontally and becoming shorter/longer vertically) and one terminate action. The details of the spatial action set A are illustrated in Fig. 4 (middle and bottom). At the $k-th$ searching step, we obtain two sequences of actions, $\mathbf{a}_k^1 = \{a_{k,1}^1, a_{k,2}^1, \ldots, a_{k,T}^1\}$ and $\mathbf{a}_k^2 = \{a_{k,1}^2, a_{k,2}^2, \ldots, a_{k,T}^2\}$, estimated from the target branch and context branch, respectively, except in the feature fusion strategy.

4) Spatial Reward: The reward function $R(s_{k,t}, a_{k,t})$ is defined as the improvement in localization accuracy after the agent selects a particular action $a_{k,t}$ at state $s_{k,t}$. The intersection-over-union (IoU) between the predicted bounding box $bb_{k,t}$ and the ground truth bg_t is used to measure the localization accuracy. The reward function is divided into two groups, namely, terminate R_T and non-terminate R_N:

$$R_N(s_{k,t}, a_{k,t}) = \lambda_1 sign(IoU(bb_{k+1,t}, bg_t) - IoU(bb_{k,t}, bg_t)) \quad (11)$$

$$R_T(s_{k,t}, a_{k,t}) = \lambda_2 sign(IoU(bb_{k,t}, bg_t) - \tau) \quad (12)$$

Intuitively, the non-terminate reward indicates that a positive reward $\lambda_1 = 1$ is received if the IoU improves when the agent performs action $a_{k,t}$ at state $s_{k,t}$ and transforms the bounding box from $bb_{k,t}$ to $bb_{k+1,t}$; otherwise, a negative reward is received. A different reward function is used for the terminate action since the bounding box is no longer changed. At the terminate state $s_{k,t}$, the agent will obtain a positive reward $\lambda_2 = 3$ when the IoU is above the given threshold τ. Similarly, the rewards of both branches are sequential, namely, $R(\mathbf{s}_k^1, \mathbf{a}_k^1) = \{R(s_{k,1}^1, a_{k,1}^1), R(s_{k,2}^1, a_{k,2}^1), \ldots, R(s_{k,T}^1, a_{k,T}^1)\}$, $R(\mathbf{s}_k^2, \mathbf{a}_k^2) = \{R(s_{k,1}^2, a_{k,1}^2), R(s_{k,2}^2, a_{k,2}^2), \ldots, R(s_{k,T}^2, a_{k,T}^2)\}$, where each element is calculated in the same way.

B. The Temporal Q-network

The temporal model is a common Q-network to find the time interval of an activity, consisting of the start frame and the end frame. We approximate the temporal Q-function using a temporal Q-network, as shown in Fig.2. The temporal Q-network consists of two FC layers of 1024 neurons each and an output layer with 6 action decisions, whose architecture is similar to that in [32].

$$Q^T(s^T, a^T) = E[\sum_{k=m}^{K} \gamma^{k-m} R(s_k, a_k) | s_k = s^T, a_k = a^T; \theta^T] \quad (13)$$

where the temporal localization agent interacts with the video environment via temporal state s^T, temporal action a^T and temporal reward $R(s^T, a^T)$. The temporal reward is calculated in a manner similar to that of the spatial reward by replacing the bounding box with the time interval in Eq.11 and Eq.12.

1) Temporal State: We introduce a deep model to capture the temporal dynamics of human activity and extract the visual temporal state. As shown in Fig.2, after computing the spatial states, a temporal pooling layer is used to generate the state for the temporal Q-network. We integrate all the temporal information within the time interval by averaging the state representations of the target branch, $s^T = avg\{s_{k,1}^1, s_{k,2}^1, \ldots, s_{k,T}^1\}$.

2) Temporal Action: As illustrated in Fig.4 (top), the action set for temporal localization is composed of 3 scaling actions (left scaling, right scaling, and centre scaling), 2 translation actions (right shifting and left shifting) and one terminate action. For example, left scaling with scaling factor α and right shifting with translation factor β are performed in the following ways:

$$t_1 = t_1, \quad t_2 = t_2 - \alpha \times (t_2 - t_1), \quad (14)$$

$$t_1 = t_1 + \beta(t_2 - t_1), \quad t_2 = t_2 + \beta(t_2 - t_1). \quad (15)$$

IV. Learning and Inference With Deep Reinforcement Learning

In this section, we first propose a unified training algorithm to learn the localization agent via deep reinforcement learning. Then, we detail the inference procedure for how to locate human activity with this trained agent.

A. Training Localization Agent via Deep Reinforcement Learning

The parameters of our localization agent are given by $\Theta = \{\theta^T, \theta^S\} = \{\theta^T, \theta^1, \theta^2\}$, and Θ is learned via the Q-learning

algorithm [31]. The complete training procedure is presented in Algorithm1. We run the agent N episodes for each video clip, where temporal localization and spatial localization are performed successively. That is, the temporal Q-network and the spatial Q-network are updated iteratively until the accurate location is reached. Specifically, the agent starts from the largest time interval $\{1, T\}$ spanning the whole video sequence and takes a temporal action during each search step to update the time interval and the temporal Q-network. θ^T is trained by gradient descent:

$$\nabla_{\theta^T} L^T = [R(s_k^T, a_k^T) + \gamma \max_{a_{k+1}^T} Q^T(s_{k+1}^T, a_{k+1}^T; \theta^T) - Q^T(s_k^T, a_k^T; \theta^T)]\nabla_{\theta^T} Q^T(s_k^T, a_k^T; \theta^T) \quad (16)$$

where $(s_k^T, a_k^T, R(s_k^T, a_k^T), s_{k+1}^T)$ is a transition sample from state s_k^T to new state s_{k+1}^T after taking action a_k^T and obtaining reward $R(s_k^T, a_k^T)$. For spatial localization, we take the 'uni-directional expert' as an example; the other methods follow a similar training procedure, with the exception that the parameters are updated according to the corresponding loss functions. Starting from the whole image $\{1, 1, W, H\}$ for each frame during the currently detected time interval, the agent takes a sequence of spatial actions to optimize the spatial Q-network. During the first several episodes $n < n_1$, we independently optimize the two branches and consider the context branch as an expert to help train the target branch. By minimizing the loss in Eq.5, we update θ^2 via deep reinforcement learning:

$$\nabla_{\theta^2} L^2 = [R(s_{k,t}^2, a_{k,t}^2) + \gamma \max_{a_{k+1,t}^2} Q^2(s_{k+1,t}^2, a_{k+1,t}^2; \theta^2) - Q^2(s_{k,t}^2, a_{k,t}^2; \theta^2)]\nabla_{\theta^2} Q^2(s_{k,t}^2, a_{k,t}^2; \theta^2) \quad (17)$$

By minimizing the loss in Eq.7, we update θ^1 via deep reinforcement learning with the Bellman equation and supervised learning with cross entropy loss:

$$\nabla_{\theta^1} L^1 = [R(s_{k,t}^1, a_{k,t}^1) + \gamma \max_{a_{k+1,t}^1} Q^1(s_{k+1,t}^1, a_{k+1,t}^1; \theta^1) - Q^1(s_{k,t}^1, a_{k,t}^1; \theta^1)]\nabla_{\theta^1} Q^1(s_{k,t}^1, a_{k,t}^1; \theta^2) - \frac{p_{k,t}^2}{p_{k,t}^1} \nabla_{\theta^1} p_{k,t}^1 \quad (18)$$

Note that the detected bounding boxes are updated in terms of only the target branch. In TBSS-DQN, each change in decision sequence is made depending on both the state of the current frame and the cues from previous frames. When it is required to change the attentional region of each frame simultaneously, TBSS-DQN succeeds in capturing temporal dependencies involving in human activity.

During training, the terminate state is reached when the IoU between the detections and ground truth exceeds the threshold τ. In the Q-learning algorithm, we adopt ε-greedy for the behaviour of the agent to avoid becoming stuck in a local optimum. In each searching step, the agent selects a random action with probability ε and takes the action depending on the previously learned policy with probability $1 - \varepsilon$. Replay memory is incorporated to store experiences and to randomly sample a mini-batch to update the ST-DQN.

Algorithm 1 Training Procedure for the ST-DQN

1: Initialize the parameters of the spatio-temporal deep Q-network $\Theta = \{\theta^1, \theta^2, \theta^T\}$;
2: Initialize the time interval $\{t_1, t_2\} = \{1, T\}$ and bounding boxes $\mathbf{bb}_1 = \{1, 1, W, H\}_{t_1}^{t_2}$
3: **for** each episode $n \in [1, N]$ **do**
4: **for** searching step $k \in [1, K]$ **do**
5: Given the current $\mathbf{bb}_k$, update the temporal DQN: $\theta^T \leftarrow \theta^T - \alpha \nabla_{\theta^T} L^T$ (Eq.16);
6: Given the current $\mathbf{bb}_k$, update the $\{t_1, t_2\}$ by a_k^T, where $a_k^T = \underset{a}{\text{argmax}}\, Q^T(s_k^T, a; \theta^T)$ (Eq.13);
7: **for** each frame $t \in [t_1, t_2]$
8: Update the context branch: $\theta^2 \leftarrow \theta^2 - \alpha \nabla_{\theta^2} L^2$ (Eq.17);
9: **if** $n < n_1$**:** Update the target branch: $\theta^1 \leftarrow \theta^1 - \alpha \nabla_{\theta^1} L^1$ (same as Eq.17);
10: **else:** Given the current θ^2, update the target branch: $\theta^1 \leftarrow \theta^1 - \alpha \nabla_{\theta^1} L^1$ (Eq.18);
11: Update the $bb_{k,t}$ by $a_{k,t}^1$, $a_{k,t}^1 = \underset{a}{\text{argmax}}\, Q^1(s_{k,t}^1, a; \theta^1)$ (same as Eq.2);
12: **end if**
13: **end for** t, k, n
14: **return** Θ;

B. Locating Activity With the Optimal Localization Policy

Once the agent is trained using the procedure described above, testing is performed in a manner similar to that of training, with the exception that the model is not updated. During inference, the localization agent runs a series of actions guided by the optimal policy, where the action with the highest predicted Q-value is selected in each step. The agent follows the optimal search path from the whole spatio-temporal space of a video sequence to the accurate activity location. After the agent reaches the terminate action, we stop the searching process and consider the last detected region to be the final proposal. Fixing the maximum number of steps is an alternative way to stop the search process. In addition to efficiency, the policy-based search is appealing because it mimics the human attention mechanism and follows the procedure of human perception. Compared with that of the two traditional fusion approaches, the computation cost of TBSS-DQN with expert is much lower since the context branch is no longer required during testing.

The Q-value is designed for proposal extraction rather than activity classification since it estimates only the improvement resulting from each change and not a discriminative score. Therefore, we adopt an external classification network, namely, two-stream inflated 3D ConvNet (I3D) [7], which is a state-of-the-art recognition model, as an evaluator to recognize the activity proposals. The I3D is pretrained on Kinetics and we replace its last layer with a new classification FC layer of $c + 1$ neurons, that is, the number of activity categories and a background class. We use the Adam optimizer [24] with a small learning rate of 1e-6 to fine-tune the parameters.

V. EXPERIMENTS

We evaluate localization performance of our proposed ST-DQN on several public human activity datasets. In this section, we first introduce the datasets, evaluation metrics and implementation details. Then, we analyse our model comprehensively. 1) We compare the frame-based and video-based localization methods. 2) We evaluate the context for activity localization. Finally, we compare to the state-of-the-art methods and present some qualitative localization results.

A. Datasets and Experimental Setup

1) Datasets: ***UCF-Sports*** *[39]:* This dataset is used for activity localization to detect the spatial locations of activities in realistic scenes. The videos in UCF-Sports are segmented into short clips, and bounding box annotations are provided for all frames. The dataset includes 150 clips from various sporting events with 10 categories of sports activities. **JHMDB and Sub-JHMDB [21], [52]**: The sub-JHMDB and JHMDB are also used for activity spatial localization. The sub-JHMDB is a subset of JHMDB where all the joints are inside the frame, and it contains 316 clips with 12 activity categories. Activity recognition for this subset is much more challenging than that of the complete JHMDB [21]. **UCF-101 [46]**: This large dataset is collected from YouTube for activity recognition with more than 13000 videos and 101 categories. For the activity localization task, detailed spatial location annotations are provided for a subset, which contains 3207 video sequences with 24 categories of activities. In contrast to UCF-Sports and sub-JHMDB, the videos in UCF-101 are relatively untrimmed, which makes the dataset more realistic and challenging for the spatio-temporal localization task. **ActivityNet-1.3 [4]** is a large dataset for activity temporal localization, including 19228 videos from 200 activity categories. The training, validation and testing sets are divided according to a 2:1:1 ratio.

2) Evaluation Metrics: A localization is accepted as correct if both the predicted activity label and the predicted location match the ground truth, that is, the classification result is accurate and the IoU with the ground truth exceeds a specified threshold δ. The IoU between two spatio-temporal paths is defined as the temporal IoU multiplied by the spatial IoU between bounding boxes averaged over all overlapping frames. To fully evaluate our ST-DQN, we consider an IoU threshold of [0.05:0.95] for spatio-temporal localization and [0.5,0.75] for temporal localization. By default, the reported metric is the mean average precision (mAP), which comprehensively represents the relationship between recall and precision, at an IoU threshold of $\delta = 0.2$. Furthermore, we report the receiver operating characteristic curves (ROCs), as in previous works [11], which are obtained by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. Unless otherwise noted, ST-DQN adopts the default fusion strategy of bidirectional experts.

3) Implementation Details: We employ the standard training and testing split for each dataset. The state generator is a multilayer CNN with the VGG-16 architecture pretrained on ImageNet. We train the ST-DQN using reinforcement learning with a ε-greedy policy for 15 epochs, each completed after performing an episode for all training videos. During the ε-greedy training, ε is annealed linearly from 1 to 0.1 over the first 10 epochs, which allows the agent to progressively utilize its own learned searching policy. Then, ε is fixed to 0.1 in the last 5 epochs, and the agent updates the ST-DQN parameters based on experience produced by its own decisions. The parameters are optimized by the Adam optimizer [24] with a learning rate of 1e-6, and dropout regularization [47] is used to avoid overfitting. For the training process, the discount factor γ in Eq.(1)(13) is set to 0.9, and we run each episode with a maximum of $K = 100$ searching steps. We utilize an experience pool of size $|D| = 1000$, and the mini-batch size is 16. By default, we use the scaling action with a factor $\alpha = 1/10$ in Eq.(14)(9), and the translation action with a factor $\beta = 1/10$ in Eq.(15). Meanwhile, the threshold τ of terminate reward in Eq.(12) is set to 0.75.

B. Frame-Based Localization vs. Video-Based Localization

One advantage of our model lies in accounting for the temporal dependencies between neighboring frames to locate activity in each frame simultaneously. We compare our video-based localization method to a frame-based localization method to verify the benefit of this sequence-to-sequence model. Specifically, the 'video-based localization method' here is equivalent to 'target branch', where only the target branch of TBSS-DQN is employed. For the 'frame-based localization method', we utilize the same temporal Q-network architecture to independently locate activity frame by frame and replace the temporal tuples (state, action, reward) with spatial tuples. As shown in Table I, we compare the mAPs at different IoU thresholds on UCF-Sports, sub-JHMDB and UCF-101. The comparisons indicate that temporal dependencies are indeed helpful for localization, which suggests that the agent makes consistent decision among neighboring frames. Since human activity changes are relatively gradual and sudden location changes usually do not occur, it is reasonable to select the same action for neighboring frames. We achieve a substantial improvement at both low IoU and high IoU by incorporating temporal relationships into the Q-network. Specifically, the mAP increases from 73.68% to 76.86%, 67.05% to 67.10% and 78.62% to 83.25% on the UCF-Sports, sub-JHMDB and UCF-101 datasets, respectively, when IoU= 0.05. Meanwhile, the change from the frame-based method to the video-based method leads to improvements of 0.59%, 1.05% and 1.14%, respectively, at the higher IoU=0.3. This result highlights the benefit of performing localization at the sequence level.

In addition, we analyse the ROC curves and precision-recall curves for the frame-based localization method and video-based localization method, as shown in Fig.5(a)-5(f), where the default IoU=0.2 is adopted. The two curves types are plotted by ranking all the activity proposals generated by the localization agent. The ROC curves and PR curves highlight the steady increase in localization performance by utilizing the sequence-to-sequence model. The reasons for the improved performance are as follows: 1) The decision consistency constraint can refine and even fix some inexact detections, which makes the model more robust. 2) Previous

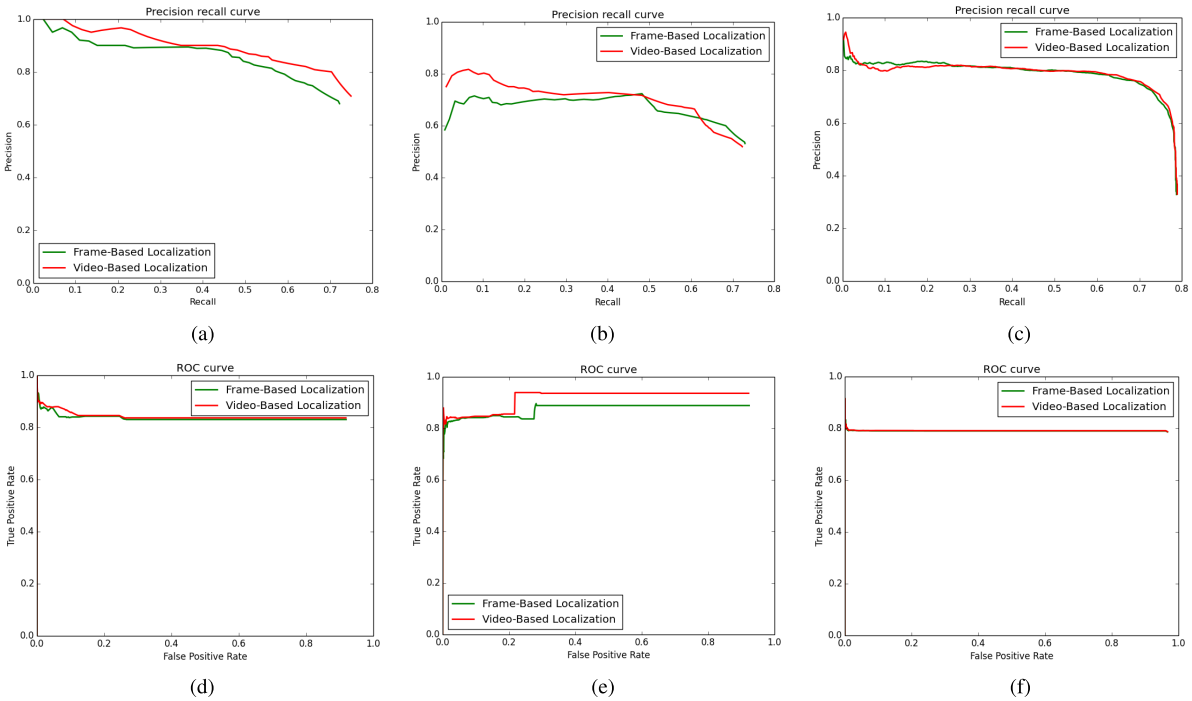


Fig. 5. Comparison of the ROC curves and precision-recall curves of the frame-based method and video-based method for activity localization at the default IoU = 0.2 on UCF-Sports, sub-JHMDB and UCF-101. The figure is best viewed in colour. (a) PR curve on UCF-Sports. (b) PR curve on sub-JHMDB. (c) PR curve on UCF-101. (d) ROC curve on UCF-Sports. (e) ROC curve on sub-JHMDB. (f) ROC curve on UCF-101.

TABLE I

COMPARE MAPS OF THE FRAME-BASED METHOD AND VIDEO-BASED METHOD FOR ACTIVITY LOCALIZATION WITH RESPECT TO DIFFERENT IOU THRESHOLDS ON UCF-SPORTS, SUB-JHMDB AND UCF-101

Methods / mAP@IoU	UCF-Sports				sub-JHMDB				UCF-101			
	0.05	0.1	0.2	0.3	0.05	0.1	0.2	0.3	0.05	0.1	0.2	0.3
Frame-based method	0.7368	0.7252	0.6863	0.6212	0.6705	0.6607	0.6174	0.4443	0.7862	0.7534	0.6680	0.5064
Video-based method	0.7686	0.7603	0.7346	0.6271	0.6710	0.6678	0.6266	0.4548	0.8325	0.7878	0.6780	0.5178

frames can provide some temporal contextual information for the current frame, such that each decision is dependent on the long-term activity sequence.

C. Evaluation of the Context for Activity Localization

Another advantage of our model is the combination of the target branch with the context branch to refine the spatio-temporal boundary of human activity. We compare the following approaches for performance evaluation to estimate the influence of context in activity localization: 1) Context Branch: employ only the context branch of TBSS-DQN; 2) Target Branch: employ only the target branch of TBSS-DQN; 3) Decision Fusion: fuse the context branch and target branch at the decision level; 4) Feature Fusion: fuse the context branch and target branch at the feature level; 5) Unidirectional Expert: consider the context branch as an expert to help train the target branch; and 6) Bidirectional Expert: the two branches supervise each other during training. The first two methods are designed to evaluate each TBSS-DQN branch separately. The last four methods aim to compare the localization performance of TBSS-DQN with those of different fusion strategies.

We follow the convention of reporting the mAP for each approach when varying the IoU threshold values in Table II. We obtain the following observations and conclusions based on these experimental results. 1) In addition to the target branch, the context branch contributes to the localization task. For instance, when utilizing only the context branch, we achieve a mAP of 44.1% at IoU=0.3 on UCF-101, which is only 7.68% less than that of the target branch. 2) The target branch and context branch are both important for activity localization, and the precision of each individual branch can be improved by incorporating both branches. For all datasets, a clear gain between the individual branch of TBSS-DQN and the complete TBSS-DQN is obtained at all IoU thresholds. Compared to the target branch, the performance of TBSS-DQN with bidirectional expert is increased by 3.16% on UCF-Sports, 9.86% on sub-JHMDB and 4.06% on UCF-101. These results highlight the benefit of combining target and context cues when performing localization. 3) The bidirectional expert is the best fusion strategy for TBSS-DQN from the perspective of both precision and speed of localization. First, TBSS-DQN with bidirectional expert achieves the highest localization precision for all datasets, *e.g.*, it outperforms TBSS-DQN with other three fusion strategies (decision fusion, feature fusion, and unidirectional expert) by 7.09%, 1.40% and 3.69%, respectively, on sub-JHMDB. Second, since TBSS-DQN with bidirectional or unidirectional expert considers the context

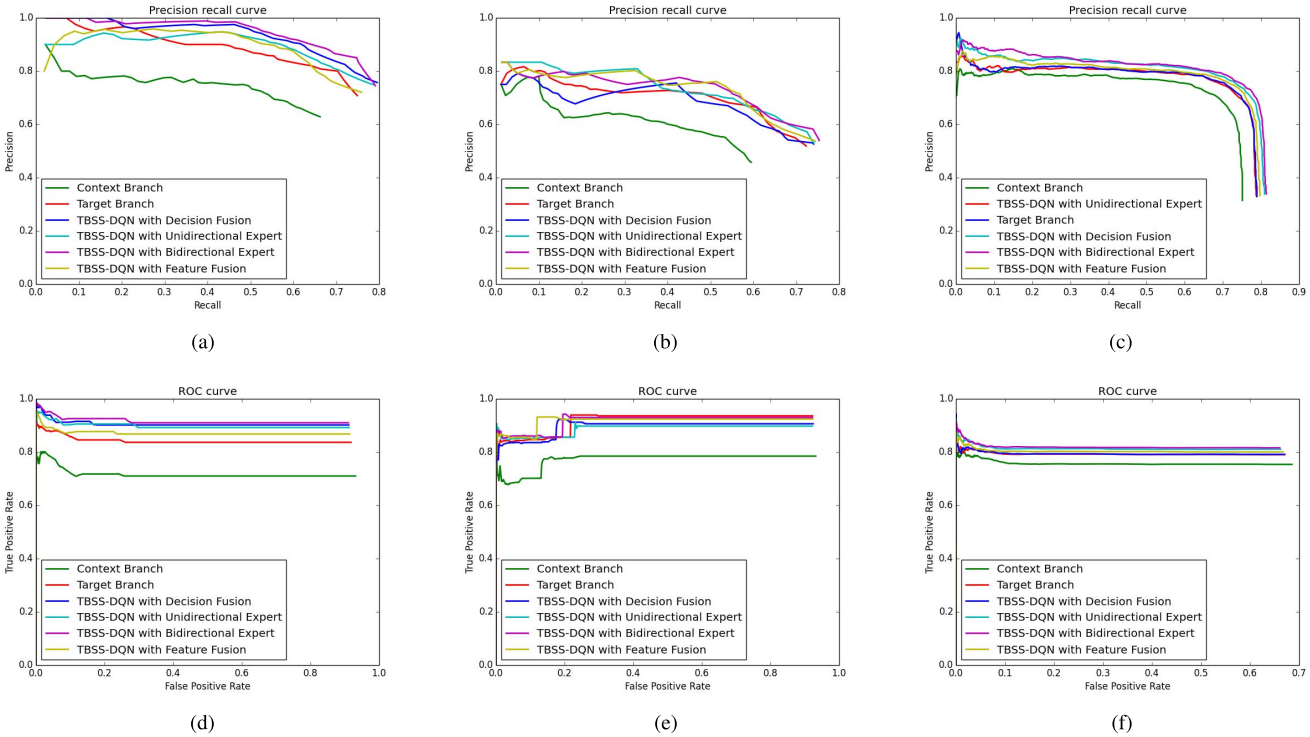


Fig. 6. Comparison of the PR curves and ROC curves of TBSS-DQN with different fusion strategies on UCF-Sports, sub-JHMDB and UCF-101. (a) PR curve on UCF-Sports. (b) PR curve on sub-JHMDB. (c) PR curve on UCF-101. (d) ROC curve on UCF-Sports. (e) ROC curve on sub-JHMDB. (f) ROC curve on UCF-101.

TABLE II

COMPARISON OF THE mAPs OF TBSS-DQN WITH DIFFERENT FUSION STRATEGIES FOR ACTIVITY LOCALIZATION WITH DIFFERENT IoU THRESHOLDS ON UCF-SPORTS, SUB-JHMDB AND UCF-101

Methods / mAP@IoU	UCF-Sports				sub-JHMDB				UCF-101			
	0.05	0.1	0.2	0.3	0.05	0.1	0.2	0.3	0.05	0.1	0.2	0.3
Context Branch	0.7352	0.7035	0.5808	0.3955	0.6379	0.6117	0.5031	0.3407	0.8245	0.7695	0.6277	0.441
Target Branch	0.7686	0.7603	0.7346	0.6271	0.6710	0.6678	0.6266	0.4548	0.8325	0.7878	0.678	0.5178
Decision Fusion	0.8016	0.7986	0.782	0.6449	0.6495	0.6495	0.643	0.4826	0.824	0.7901	0.7061	0.5531
Feature Fusion	0.7765	0.7765	0.7379	0.6506	0.6915	0.6841	0.66	0.5395	0.8211	0.7859	0.6858	0.5187
Unidirectional Expert	0.8013	0.7904	0.7647	0.6449	0.6725	0.6681	0.6533	0.5166	0.8268	0.7871	0.6828	0.5251
Bidirectional Expert	0.8019	0.7941	0.7829	0.6587	0.6922	0.685	0.6738	0.5535	0.8297	0.7975	0.7131	0.5584

branch as an expert only to help train the target branch, which is not necessary during testing, its localization speed is faster than that of TBSS-DQN with decision fusion or feature fusion.

To comprehensively validate the importance of context for localization, the PR curves and ROC curves are drawn to compare TBSS-DQN with various fusion strategies in Fig.6(a)-6(f). Two interesting points can be noted in these graphs. First, TBSS-DQN with bidirectional expert obtains consistent results for all datasets, while the results of the other three fusion methods vary among datasets. In contrast to TBSS-DQN with bidirectional expert, whose PR and ROC curves are always the highest, the curve of TBSS-DQN with feature fusion is higher than that of TBSS-DQN with decision fusion on UCF-Sports and sub-JHMDB, while the opposite is observed for UCF-101. Second, when the recall value increases, the PR curves of the four fusion methods all gradually surpass that of the target branch. This fact clearly demonstrates that combining context results in more accurate localization for difficult examples.

D. Reward and Action Selection

We also perform ablation studies to more comprehensively evaluate our model. The scaling factor α, translation factor β, terminate reward λ_2 and action type influence both the precision and localization speed. Therefore, we measure the impact of these factors on the final localization performance by running ST-DQN with different action selections and reward selections on JHMDB. The action selection is in terms of the scaling factor (translation factor) and the action type, where we set $\alpha = \beta$, since this parameter determines the change scale regardless of changes in scaling and translation. The reward selection is determined by the terminate reward λ_2, which balances the terminate state and non-terminate state.

As illustrated in Table III, the mAP is a measure of a model's localization performance, and the search step (namely, the proposal number in this paper) is a measure of the localization speed. Additionally, we report the processing speed during the testing stage for a server with a 24-core XEON processor

TABLE III
THE LOCALIZATION RESULTS AND DETECTION SPEED (ACTION TYPE, α, λ_2)

Methods	[34]	[40]	[45]	(9,1/10,3)	(14,1/10,3)	(6,1/6,3)	(6,1/12,3)	(6,1/10,3)	(6,1/10,1)	(6,1/10,5)
mAP@0.2	74.3%	72.63%	73.8%	79.07%	79.21%	72.15%	80.48%	79.63%	**80.92**%	76.43%
mAP@0.5	73.09%	71.50%	72.0%	77.78%	**78.47**%	66.57%	78.37%	77.50%	75.56%	74.87%
mAP@0.75	-	43.3%	44.5%	**51.83**%	50.79%	23.87%	31.0%	29.21%	28.09%	26.26%
Proposal No.	256	256	10*c	41.78	48.41	7.187	12.44	10.21	10.45	11.43
Overall speed (fps)	4	4	7	2.979	1.994	9.559	5.329	7.134	7.127	6.377

TABLE IV
TEMPORAL LOCALIZATION RESULTS OF OUR TEMPORAL DQN

Methods/IoU	UCF-101		
	mAP@0.5	mAP@0.75	Proposal No.
Temporal DQN	71.85%	57.58%	3.23

Methods/IoU	ActivityNet		
	mAP@0.5	mAP@0.75	Proposal No.
Wang *et al.* [55]	42.28%	3.76%	-
SCC [18]	40.0%	17.90%	100
SSN [61]	39.12%	23.48%	56
CDC [42]	43.83%	25.88%	11.2
Lin@5 [28]	42.57%	28.26%	5
Lin@100 [28]	44.39%	**29.65%**	100
Temporal DQN	**45.85%**	28.08%	4.67

with 32G RAM and an NVIDIA 2080Ti GPU. We compare the model with different action sets, which include a 6-action set, a 9-action set and a 14-action set consisting of the former two sets. The results show the following: 1) As the value of the scaling factor decreases, the mAP and the required number of search steps both increase. 2) Since the length-width ratio is fixed for 6 actions, utilizing 14 actions or 9 actions results in a more accurate localization boundary, especially for higher IoU threshold (*e*.g., IoU=0.75), but it requires many more steps to locate the target activity. In fact, utilizing 6 actions is the simplest means to transform the bounding box to cover any location of an image, without any redundant actions, which means the different actions result in the same IoU value or reward. The redundant actions and reciprocating actions (*e.g.,* moving left/ right) are the main reasons why utilizing 14 actions or 9 actions requires more steps for convergence. Note that the localization result of ST-DQN with 6 actions is on par with or even better than ST-DQN with 9 actions or 14 actions using fewer proposals (approximately 10 *vs.* approximately 40) when the IoU threshold is less than or equal to 0.5. The results demonstrate that the ST-DQN with 6 actions is feasible when the precision requirement is not rigid, where the model can locate activity faster under the condition that a certain precision is assured, and ST-DQN with 9 actions or 14 actions is more suitable for pursuing higher precision. Utilizing 9 actions, 14 actions or a small scaling factor makes the search process slower and more refined. 3) When too small of a terminate reward is adopted, the terminate state and non-terminate state are treated equally, while too large of a terminate reward makes the agent emphasize the terminate state and neglect the intermediate process, both of which decrease the localization performance. To balance the localization precision with the convergence speed, we adopt the 6-action set (with $\alpha = 1/10$ and $\lambda_2 = 3$) by default in this paper, which results in satisfactory localization precision with fewer steps. 4) Clearly, as the number of steps to convergence increases, the fps decreases, and the testing speed slows.

E. Comparison to the State-of-the-Art

We continue the evaluation with a comparison to the recently proposed state-of-the-art methods in Table V. In Table IV, we report the mAP of the temporal localization obtained by our temporal DQN on UCF-101 and ActivityNet.

First, we compare our ST-DQN with frame-based localization methods [16], [23], [40], [45], [58], which independently detect activity for each frame (or several successive frames) and then perform video-level detection via temporal post-processing. Most of the compared methods are extensions of faster R-CNN [8], [34], [40], which is currently the best object detection method. Specifically, we achieve clear improvements of 15.78% and 6% at IoU = 0.5 on UCF-101 and JHMDB, respectively, compared to the method in [40], which is also a two-stream architecture consisting of an appearance network and motion network; we outperform the method in [8], which is a generalization of R-CNN from 2D to 3D, by a margin of 0.6% and obtain a mAP of 77.5% on JHMDB. Although [23], [45] achieves a better result at IoU = 0.2 on UCF-101, we achieve the highest mAP of 51.64% at IoU = 0.5. Moreover, our ST-DQN achieves comparable and even better performance when using only RGB information or fewer proposals, where c denotes the number of activity categories.

Then, we compare our method to some video-based localization methods [13], [63] that directly extract video-level proposals in the spatio-temporal domain. At a threshold IoU = 0.2, on UCF-101, we obtain a mAP of 71.31%, compared to the 42.8% reported by [63], which considers spatio-temporal video tubes as activity proposals. Similarly, our ST-DQN substantially outperforms [13] by 36.81%, where activity proposals are extracted from dense trajectories. Although the precision of ST-DQN with 6 actions is lower at a higher IoU due to its fixed length-width ratio, that of ST-DQN with 9 actions is on par with or even better than the state-of-the-art methods [23] at IoU=0.75 and IoU=0.5:0.95. In the case of low IoU threshold (*e*.g., less than or equal to 0.5), ST-DQN with 6 actions is the optimal choice to fast locate activity, where it achieves the best performance and is even better than ST-DQN with 9 actions. For temporal localization, our temporal DQN outperforms [18], [42], [61] on ActivityNet at both IoU = 0.5 and IoU = 0.75. Compared to [28], which is a typical two-stream model, we obtain a higher mAP at IoU = 0.5 and comparable performance with a similar proposal number at IoU = 0.75.

Overall, our ST-DQN outperforms the state-of-the-art methods on JHMDB and is on par with [23] and [28] on

TABLE V
COMPARISON TO THE STATE-OF-THE-ART METHODS FOR ACTIVITY LOCALIZATION

Methods/IoU	Proposal	Modality	JHMDB				UCF-101				
			0.2	0.5	0.75	0.5:0.95	0.05	0.2	0.5	0.75	0.5:0.95
Yu *et al.* [63]	10K	-	-	-	-	-	49.9%	42.8%	-	-	-
Hou *et al.* [8]	256	RGB Stream	78.4%	76.9%	-	-	54.7%	47.1%	-	-	-
Gemert *et al.* [13]	2299	-	-	-	-	-		34.5%		-	-
Kalogeiton *et al.* [23]	10*c	RGB Stream	74.2%	73.7%	**52.1%**	44.8%	-	**77.2%**	51.4%	22.7%	25.0%
Saha (RGB) *et al.* [40]	256	RGB Stream	52.94%	51.34%	-	-	68.74%	56.91%	30.67%	-	-
Saha *et al.* [40]	256	RGB + OF	72.63%	71.50%	43.3%	40.0%	79.12%	66.75%	35.86%	7.9%	14.4%
Gkioxari *et al.* [16]	2K	RGB + OF	53.3%	-	-	-	-	-	-	-	-
Weinzaepfel *et al.* [58]	256	RGB + OF	63.1%	60.7%	-	-	54.28%	46.77%	-	-	-
Mettes *et al.* [30]	1449/6706	-		-	-	-	-	34.8%	-	-	-
Weinzaepfel *et al.* [59]	256	RGB Stream	-	63.9%	-	-	70.0%	57.4%	-	-	-
Singh (RGB) *et al.* [45]	10*c	RGB Stream	60.8%	59.7%	-	-	-	69.8%	40.9%	-	-
Singh *et al.* [45]	10*c	RGB + OF	73.8%	72.0%	44.5%	41.6%	-	73.5%	46.3%	15.0%	20.4%
Peng *et al.* [34]	256	RGB + OF	74.3%	73.09%	-	-	78.76%	72.86%	32.1%	2.70%	7.30%
ST-DQN (6 actions)	Approx. 10	RGB Stream	**79.63%**	77.50%	29.21%	26.26%	**82.97%**	71.31%	**51.64%**	8.834%	17.94%
ST-DQN (9 actions)	Approx. 40	RGB Stream	79.07%	**77.78%**	51.83%	**45.24%**	82.45%	68.55%	51.03%	**23.09%**	**25.45%**

Fig. 7. Qualitative activity localization results on UCF-101. Comparisons between ST-DQN with 6 actions (first row), the target branch (second row) and the frame-based method (third row). Comparisons between ST-DQN with 6 actions (fourth row and sixth row) and ST-DQN with 9 actions (fifth row and seventh row). Green boxes indicate our detections and red boxes denote the ground truth.

UCF-101 and ActivityNet, respectively. Moreover, the promising localization mAP of our ST-DQN is achieved by utilizing much fewer proposals (*e.g.,* approx. 10/40 *vs.* 10K [63], approx. 10/40 *vs.* 256 [34], [40], [59]). Note that we consider the average of the convergence steps to be the proposal number, which means that the agent needs to generate approximately 10/40 and 3-4 proposals before reaching the target location for spatio-temporal and temporal localization, respectively. We detail these characteristics in Section V.D and Fig.8.In the 'proposal + classification' framework [59], [63], the time complexity is proportional to the number of proposals N, namely, $O(N \cdot [O(NN_{prop}) + O(NN_{class})])$, which refer to the time complexity of proposal network and classification network, respectively. The complexity of the proposal network

TABLE VI

COMPARISON TO THE STATE-OF-THE-ART METHODS FOR ACTIVITY PROPOSAL

Methods	MABO	Recall	Proposal No.
Gemert *et al*. [13]	46.79	46.38	2299
Li *et al*. [27]	40.84	39.64	18
Hongyuan *et al*.(RGB) [71]	53.30	59.59	30
Hongyuan *et al*. [71]	**54.93**	61.42	30
Our ST-DQN	54.88	**61.50**	Approx. 10

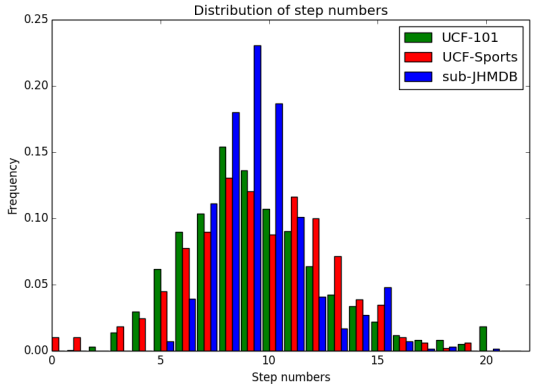


Fig. 8. The distributions of the number of steps necessary to locate target activity on UCF-Sports, sub-JHMDB and UCF-101. Almost all the detections require fewer than 20 steps, with an average of approximately 10 steps.

is lower than that of most state-of-the-art activity classification models. For example, in this paper, the number of parameters, which is commonly used for measuring the complexity of deep model, is 23.9M in ST-DQN, while that of I3D is 25M. It clearly demonstrates the superior ability of ST-DQN to extract proposals effectively and efficiently and the benefit of explicitly integrating context information and temporal dependencies.

F. Proposal Quality Evaluation

The proposed ST-DQN focuses mainly on spatio-temporal activity proposal generation, so we also compare the proposed method with some recently reported proposal methods [13], [27], [71] on the UCF-101 dataset, as shown in Table VI. Several common proposal evaluation metrics are used, including mean average best overlap over all classes (MABO), recall and average number of proposals, where recall is computed at an IoU threshold of 0.5. Our ST-DQN obtains the highest recall with approximately 10 proposals, which demonstrates that our ST-DQN can produce proposals with good quality for further activity localization. Moreover, we achieve a MABO comparable to that of Zhu *et al*. [71] with fewer proposals, and we achieve much better performance in terms of both recall and MABO when using only the RGB stream.

G. Post-Processing Evaluation

In ST-DQN, we consider the last extracted proposal as the final detection. However, there may be an alternative when it is desired to perform some post-processing to fine-tune these proposals, such as the bounding box regression. Following the common steps, we parameterize the transformation in terms of four functions, including the scale-invariant translation of x_1 and y_1: $d_x(bb) = (bg[0] - bb[0])/(bb[2] - bb[0])$, $d_y(bb) = (bg[1] - bb[1])/(bb[3] - bb[1])$ and the log-space translation of the width and height: $d_w(bb) = \log[(bg[2] - bg[0])/(bb[2] - bb[0])]$, $d_h(bb) = \log[(bg[3] - bg[1])/(bb[3] - bb[1])]$. Each function is modelled by a linear function of the state of bb, and its parameters are optimized by minimizing $\sum_i (t_i^* - w^T s(bb))^2$, where t_i^* is the regression target. For more details, please refer to [14]. In Table VII, we compare the localization performance between ST-DQN without and with box regression (BR) on sub-JHMDB. The results demonstrate that the bounding box regression indeed improves the precision of proposals for both ST-DQN with 6 actions and ST-DQN with 9 actions, especially for those proposals with high IoU, since the linear assumption is tenable only if the detection is close to the ground truth. Compared to ST-DQN with 6 actions, the effect of regression is a little more remarkable for ST-DQN with 9 actions, since ST-DQN with 9 actions can achieve better localization performance at high IoU.

TABLE VII

COMPARISON OF THE LOCALIZATION PERFORMANCE BETWEEN ST-DQN WITHOUT AND WITH BOX REGRESSION (BR) ON SUB-JHMDB

Methods/IoU	@0.1	@0.2	@0.3	@0.5	@0.75
ST-DQN(6 actions)	0.6850	0.6738	0.5535	0.5014	0.2152
ST-DQN(6 actions)+BR	0.7064	**0.6812**	0.5719	0.5435	0.2362
ST-DQN(9 actions)	0.680	0.6626	0.6276	0.5407	0.3814
ST-DQN(9 actions)+BR	**0.7078**	0.6798	**0.6443**	**0.5801**	**0.4122**

TABLE VIII

THE EFFECT OF THE INTERMEDIATE ACTIONS FOR ST-DQN WITH 9 ACTIONS ON SUB-JHMDB

Step Number/IoU	mAP@0.2	mAP@0.5	mAP@0.75
Step=1	0.1426	0.0294	0
Step=5	0.3628	0.1318	0.0205
Step=10	0.4944	0.3428	0.1091
Step=20	0.6218	0.4213	0.2753
Step=30	0.6306	0.5144	0.3774
Step=40	0.6577	0.5379	**0.3875**
Step=50	0.6623	0.5379	0.3809
Step=100	**0.6626**	**0.5407**	0.3814

TABLE IX

THE EFFECT OF THE INTERMEDIATE ACTIONS FOR ST-DQN WITH 6 ACTIONS ON SUB-JHMDB

Step Number/IoU	mAP@0.2	mAP@0.5	mAP@0.75
Step=1	0.1426	0.0294	0
Step=3	0.3875	0.1264	0.0393
Step=5	0.5013	0.2809	0.0927
Step=8	0.6227	0.4795	0.1404
Step=10	0.6242	0.4916	0.1917
Step=12	0.6523	**0.5056**	0.2124
Step=20	**0.6738**	0.5014	**0.2152**

H. Qualitative Analysis

In order to present the effect of intermediate actions, we report the variation of localization precision with search step as shown in Table VIII and Table IX for ST-DQN with 9 actions and ST-DQN with 6 actions, respectively. Clearly, for both action sets, as the search step increases (namely, taking more actions), the mAP increases correspondingly,

which demonstrates that our ST-DQN indeed increases IoU step by step and locates activity iteratively. Additionally, it observes that there exists some outliers, such as the mAP at step=40 and IoU=0.75 in Table VIII and the mAP at step=12 and IoU=0.5 in Table IX. For the outlier in Table IX, the reason is that the model is to pursue a higher IoU and cannot successfully converge to a terminate status. The reciprocating actions in 9-action set result in the outlier in Table VIII. Compared to the model with 9 actions, once the model with 6 actions misses the target activity, there is no chance to make it up, but these reciprocating actions in 9-action set result in the jitter of bounding box around the target activity.

Finally, for a more intuitive and clearer explanation, we present some qualitative results in Fig.7. 1) Comparisons between ST-DQN with 6 actions, the target branch and the frame-based method. Our ST-DQN can generate proposals with high precision, not only for activities without movement but also for moving activities, such as cliff diving and skiing. As shown in the first three rows, we provide some qualitative examples comparing the performance of ST-DQN with and without contextual/temporal information. Compared to the target branch and frame-based method, incorporating context and temporal relation leads to fewer missed detections and a more accurate boundary when the activity location in one frame is vague or occluded. For instance, in the first row, we reach the correct location, whereas in the second and third rows, there is a large variation in the detections. 2) Comparisons between ST-DQN with 6 actions and ST-DQN with 9 actions. As shown in the last four rows, ST-DQN with both action sets can locate the target activity, while the detected boundary of ST-DQN with 6 actions is less accurate, since its fixed length-width ratio. Furthermore, our ST-DQN can efficiently achieve promising localization with a much smaller number of activity proposals, where the distribution of the number of actions required by the agent to reach the target location is plotted in Fig.8. The distribution has an average of 10, and more than 90% of the detections require fewer than 15 steps to terminate.

VI. Conclusion

In this paper, we propose a unified spatio-temporal deep Q-network to effectively and efficiently extract activity proposals. Following the process of human perception, we use a coarse-to-fine searching strategy and train a localization agent to locate activity gradually and iteratively, with the ability to decide where to focus attention next based on predefined actions. We reduce the required number of proposal candidates by casting the activity localization problem as a Markov decision process and searching the target activity according to the learned spatio-temporal policy. We achieve better performance than frame-based localization methods by introducing the sequence-to-sequence Q-network into the activity localization process with consideration of the temporal interdependency of neighboring frames. By incorporating the context branch and target branch to exploit contextual relations and provide double cues for localization, we obtain a more accurate spatio-temporal boundary.

Acknowledgment

Part of this work was done when the first author visited RPI. The authors thank professor Qiang Ji for his help and advice.

References

[1] M. S. Aliakbarian, F. Saleh, B. Fernando, *et al.*. "Deep action- and context-aware sequence learning for activity recognition and anticipation," 2016, *arXiv:1611.05520*. [Online]. Available: https://arxiv.org/abs/1611.05520

[2] M. Bellver, X. Giró-i-Nieto, F. Marqués, and J. Torres, "Hierarchical object detection with deep reinforcement learning," 2016, *arXiv:1611.03718*. [Online]. Available: https://arxiv.org/abs/1611.03718

[3] F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1914–1923.

[4] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 961–970.

[5] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2015, pp. 2488–2496.

[6] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li, "Attention-aware face hallucination via deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1656–1664.

[7] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4724–4733.

[8] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5823–5832.

[9] J. Choi, J. Kwon, and K. M. Lee, "Real-time visual tracking by deep reinforced decision making," 2017, *arXiv:1702.06291*. [Online]. Available: https://arxiv.org/abs/1702.06291

[10] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen, "Temporal context network for activity localization in videos," 2017, *arXiv:1708.02349*. [Online]. Available: https://arxiv.org/abs/1708.02349

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[12] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2782–2795, Nov. 2013.

[13] J. C. van Gemert, M. Jain, E. Gati, and C. G. M. Snoek, "APT: Action localization proposals from dense trajectories," in *Proc. BMVC*, 2015, pp. 1–13.

[14] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[15] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with R*CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1080–1088.

[16] G. Gkioxari and J. Malik, "Finding action tubes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 759–768.

[17] M. Hasan and A. K. Roy-Chowdhury, "Context aware active learning of activity recognition models," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4543–4551.

[18] F. C. Heilbron, W. Barrios, V. Escorcia, and B. Ghanem, "SCC: Semantic context cascade for efficient action detection," in *Proc. CVPR*, Jul. 2017, pp. 3175–3184.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. M. Snoek, "Action localization with tubelets from motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 740–747.

[21] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3192–3199.

[22] Z. Jie, X. Liang, J. Feng, X. Jin, W. Lu, and S. Yan, "Tree-structured reinforcement learning for sequential object localization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 127–135.

[23] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Action tubelet detector for spatio-temporal action localization," 2017, *arXiv:1705.01861*. [Online]. Available: https://arxiv.org/abs/1705.01861

[24] D. P. Kingma, and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–13.

[25] X. Kong, B. Xin, Y. Wang, and G. Hua, "Collaborative deep reinforcement learning for joint object search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7072–7081.

[26] T. Lan, Y. Wang, and G. Mori, "Discriminative figure-centric models for joint action localization and recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2003–2010.

[27] N. Li, D. Xu, Z. Ying, Z. Li, and G. Li, "Searching action proposals via spatial actionness estimation and temporal path inference and tracking," in *Proc. Asian Conf. Comput. Vis.* Springer, 2016, pp. 384–399.

[28] T. Lin, X. Zhao, and Z. Shou, "Temporal convolution based action proposal: Submission to activitynet 2017," 2017, *arXiv:1707.06750*. [Online]. Available: https://arxiv.org/abs/1707.06750

[29] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff, "Action recognition and localization by hierarchical space-time segments," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2744–2751.

[30] P. Mettes, J. C. van Gemert, and C. G. M. Snoek, "Spot on: Action localization from pointly-supervised proposals," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 437–453.

[31] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: https://arxiv.org/abs/1312.5602

[32] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[33] A. Nair *et al.*, "Massively parallel methods for deep reinforcement learning," 2015, *arXiv:1507.04296*. [Online]. Available: https://arxiv.org/abs/1507.04296

[34] X. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 744–759.

[35] V. Ramanathan, K. Tang, G. Mori, and L. Fei-Fei, "Learning temporal embeddings for complex video analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4471–4479.

[36] L. Shao, S. Jones, and X. Li, "Efficient search and localization of human actions in video databases," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 504–512, Mar. 2014.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[39] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2008, pp. 1–8.

[40] S. Saha, G. Singh, M. Sapienza, P. H. S. Torr, and F. Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 58.1–58.13.

[41] W. Choi, K. Shahid, and S. Savarese, "Learning context for collective activity recognition," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3273–3280.

[42] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1417–1426.

[43] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1049–1058.

[44] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 1, no. 4, pp. 568–576.

[45] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin, "Online real-time multiple spatiotemporal action localisation and prediction," 2016, *arXiv:1611.08563*. [Online]. Available: https://arxiv.org/abs/1611.08563

[46] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*. [Online]. Available: https://arxiv.org/abs/1212.0402

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[48] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2642–2649.

[49] D. Tran and J. Yuan, "Max-margin structured output regression for spatio-temporal action localization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 350–358.

[50] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.

[51] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2014, pp. 3551–3558.

[52] L. Wang, Y. Qiao, and X. Tang, "Video action detection with relational dynamic-poselets," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 565–580.

[53] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4305–4314.

[54] M. Wang, B. Ni, and X. Yang, "Recurrent modeling of interaction context for collective activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7408–7416.

[55] R. Wang and D. Tao, "UTS at activitynet 2016," in *Proc. AcitivityNet Large Scale Activity Recognit. Challenge*, no. 8, 2016.

[56] T. Wang, S. Wang, and X. Ding, "Detecting human action as the spatio-temporal tube of maximum mutual information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 277–290, Feb. 2014.

[57] X. Wang and Q. Ji, "Hierarchical context modeling for video event recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1770–1782, Sep. 2017.

[58] P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Learning to track for spatio-temporal action localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3164–3172.

[59] P. Weinzaepfel, X. Martin, and C. Schmid, "Human action localization with sparse spatial supervision," 2016, *arXiv:1605.05197*. [Online]. Available: https://arxiv.org/abs/1605.05197

[60] B. Wu, C. Yuan, and W. Hu, "Human action recognition based on context-dependent graph kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2609–2616.

[61] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang, "A pursuit of temporal accuracy in general activity detection," 2017, *arXiv:1703.02716*. [Online]. Available: https://arxiv.org/abs/1703.02716

[62] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2678–2687.

[63] G. Yu and J. Yuan, "Fast action proposals for human action detection and search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1302–1311.

[64] G. Yu, J. Yuan, and Z. Liu, "Propagative Hough voting for human activity detection and recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 1, pp. 87–98, Jan. 2015.

[65] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2442–2449.

[66] J. Yuan, Z. Lin, and Y. Wu, "Discriminative video pattern search for efficient action detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1728–1743, Sep. 2011.

[67] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1349–1358.

[68] D. Zhang, H. Maei, X. Wang X, and F. Wang, "Deep reinforcement learning for visual object tracking in videos," 2017, *arXiv:1701.08936*. [Online]. Available: https://arxiv.org/abs/1701.08936

[69] Y. Zhang, Y. Zhang, E. Swears, N. Larios, Z. Wang, and Q. Ji, "Modeling temporal interactions with interval temporal Bayesian networks for complex activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2468–2483, Oct. 2013.

[70] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," 2017, *arXiv:1704.06228*. [Online]. Available: https://arxiv.org/abs/1704.06228

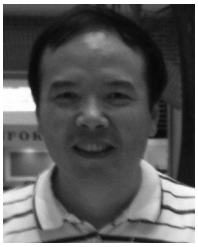
[71] H. Zhu, R. Vial, and S. Lu, "TORNADO: A spatio-temporal convolutional regression network for video action proposal," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5814–5822.

Wanru Xu received the B.S. degree in biomedical engineering and the Ph.D. degree in signal and information processing from Beijing Jiaotong University, Beijing, China, in 2011 and 2017, respectively. She is currently a Post-Doctoral Researcher with the School of Computer and Information Technology, Beijing Jiaotong University. Her current research interests include computer vision, machine learning, and pattern recognition.

Jian Yu received the B.S. and M.S. degrees in mathematics and the Ph.D. degree in applied mathematics from Peking University, Beijing, China, in 1991, 1994, and 2000, respectively. He is currently a Professor with Beijing Jiaotong University, Beijing, and the Director of the Beijing Key Laboratory of Traffic Data Analysis and Mining. His current research interests include machine learning, image processing, and pattern recognition.

Zhenjiang Miao (M'11) received the B.E. degree from Tsinghua University, Beijing, China, in 1987, and the M.E. and Ph.D. degrees from Northern Jiaotong University, Beijing, in 1990 and 1994, respectively. From 1995 to 1998, he was a Post-Doctoral Fellow with the Ecole Nationale supérieure d'électrotechnique, d'électronique, d'informatique, d'hydraulique et des télécommunications, Institut National Polytechnique de Toulouse, Toulouse, France, and was a Researcher with the Institute National de la Recherche Agronomique, Sophia Antipolis, France. From 1998 to 2004, he was with the Institute of Information Technology, National Research Council Canada, Nortel Networks, Ottawa, Canada. He joined Beijing Jiaotong University, Beijing, in 2004. He is currently a Professor and the Director of the Media Computing Center, Beijing Jiaotong University, and the Director of the Institute for Digital Culture Research, Center for Ethnic and Folk Literature and Art Development, Ministry Of Culture, China. His current research interests include image and video processing, multimedia processing, and intelligent human–machine interaction.

Lili Wan received the Ph.D. degree in computer application from Beihang University in 2007. From 2014 to 2015, she was a Visiting Researcher with the GrUVi Lab, Simon Fraser University (SFU), Canada. She is currently an Associate Professor with the Institute of Information Science, Beijing Jiaotong University, China. Her current research interests are shape analysis, 3D vision, VR, and AR.



Qiang Ji (F'15) received the Ph.D. degree from the University of Washington. He is currently a Professor with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute (RPI). From 2009 to 2010, he served as a Program Director of the National Science Foundation, managing NSF's machine learning and computer vision programs. Prior to joining RPI in 2001, he was an Assistant Professor with the Department of Computer Science, University of Nevada, Reno. He also held research and visiting positions with the Beckman Institute, University of Illinois at Urbana–Champaign, the Robotics Institute, Carnegie Mellon University, and the U.S. Air Force Research Laboratory. He is a fellow of the IAPR.